

SOPHISM: AN END-TO-END SOFTWARE INSTRUMENT SIMULATOR

J. BLANCO RODRÍGUEZ

GACE-UV, Catedrático José Beltrán 2, 46090 Paterna-Valencia, Spain

AND

J.C. DEL TORO INIESTA, D. OROZCO SUÁREZ

Instituto de Astrofísica de Andalucía (CSIC), Apdo. de Correos 3004, E-18080 Granada, Spain, jti@iaa.es

AND

V. MARTÍNEZ PILLET

National Solar Observatory, ...

AND

J. HIRZBERGER, A. FELLER, T. RIETHMÜLLER, A. LAGG

MPS...

Submitted to ApJSS

ABSTRACT

Subject headings: Sun: magnetic fields – Sun: photosphere – Sun: polarimetry – Sun: spectroscopy – Instrumentation

1. INTRODUCTION

Extracting information from astrophysical objects is an increasingly demanding task that requires more and more sophisticated instruments that set limiting conditions in terms of spatial, temporal, spectral, and polarimetric resolution, sensitivity, and accuracy. Designing a new instrument is, therefore, a challenging endeavor including many different disciplines that must be combined and traded off so that fulfillment of a number of specifications, derived from the scientific requirements, is reached. A systems engineering view (a global view of the system and all its inter-related sub-systems) is thus mandatory during the design and fabrication phases in order to assess the instrument performance and behavior. We can even acknowledge that this global system understanding is useful during the operation periods in order to diagnose possible errors, malfunctions or mistakes. This turns out to be crucial when one is dealing with a space-borne instrument and no direct manipulation is allowed during operation.

Modern instruments are complicated combinations of optics, mechanics, electronics, and software, all of which should be taken into account for the final result. The recent advent in solar physics of realistic magnetohydrodynamic (MHD) simulations of the photosphere like ... **CITATIONS** ... opens the possibility to include everything from the Sun. An end-to-end simulator of the instrument, from the object's light input up to the detector output and data storage, is then highly appreciable. SOPHISM (SO/PHI Software Simulator) is such a type of simulator that is specifically designed for the Polarimetric and Helioseismic Instrument (PHI or SO/PHI; Solanki et al. 2015) for the ESA-NASA's *Solar Orbiter* mission. Although specifically conceived for a given instrument, its modular structure and the general character of the instrument sub-systems may make it useful *mutatis mutandis* for a number of other instruments. It is written in IDL language so that it is easily accessible to scientists who can modify given modules in order to cope with different specifications or even different sub-systems.

The concept and main description of the simulator are presented in Section 2. Section 3 deals with the actual implementation and structure of the simulator, describing the different

processes that are taken into account. Section 4, shows examples of outputs from each part of the simulator and an analysis case. Finally, small plans for the future steps are presented in Section 5.

2. CONCEPT

As its name indicates, SO/PHI is a polarimetric and helioseismic imager. Hence, it is thought to image the Sun, to measure the solar vector magnetic fields (through polarimetry), and to study different modes of solar oscillations (through spectroscopy). Imaging is made by means of two telescopes that alternatively map either the entire Sun —with the so-called Full Disk Telescope (FDT)— or a highly resolved zone of it —with the so-called High Resolution Telescope (HRT). The two telescopes can be switched by a feed selection mechanism, depending on the different modes of operation. In more detail, the FDT has a 17.5 mm diameter aperture, with **a F-number of 33.5 and a FOV of 120x120 arcmin. On the other hand, the HRT presents an aperture of 140 mm in diameter, approximately 30 of F-number and 17x17 arcmin FOV.** Polarimetry is made with the help of a pair of nematic liquid crystal variable retarders (LCVRs) as the modulator and a linear polarizer as the analyzer. Only four polarization measurements ($N_p = 4$) are needed to obtain all four Stokes parameters, although observation modes with only two measurements ($N_p = 2$; $I \pm V$) are also foreseen. The system is aimed at reaching nominal polarimetric accuracies (rms value at the Stokes Q , U , and V continuum over the average Stokes I continuum) of 10^{-3} . This is achieved by accumulating N_a polarization filtergrams before polarimetric demodulation. Spectroscopy is made through a 0.3 nm wide pre-filter and a solid, LiNbO₃ Fabry-Pérot etalon with a FWHM of 8.5 pm. Since some residual jittering may not be fully corrected by the spacecraft, a correlation tracker is included that attenuates those perturbing motions, ensures the pointing stability and, hence, the image quality. We refer the interested reader to Solanki et al. (2015) for the general description of the instrument and to Martínez Pillet (2007), Del Toro Iniesta and Martínez Pillet (2012), and [HELIOSEISMIC PAPER] for further details into the magnetographic and tachographic performance of the instrument.

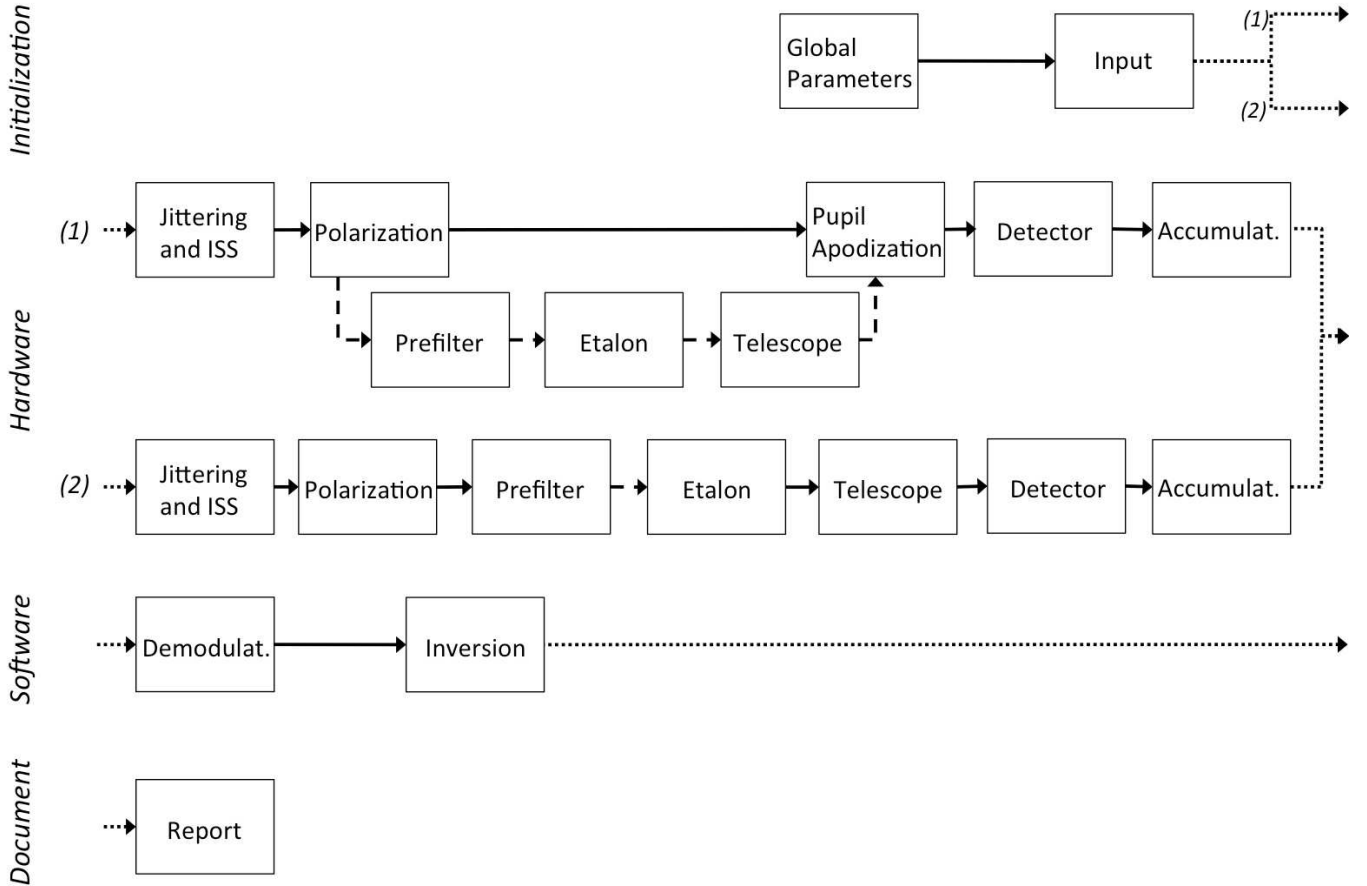


FIG. 1.— Block diagram of the simulator. See text for an explanation of the different modules.

The simulations made by SOPHISM help decide on functional and observational parameters that need be used during the mission along with error tolerances and end-of-life expectations, which are crucial for a space-borne instrument like SO/PHI.

SOPHISM is made up of blocks or modules, each one dealing with a particular effect or sub-system simulation. Inputs and outputs to each block are standard so that one can make a choice on the number of modules to be used. This is particularly useful when trying to understand the isolated consequences of a given sub-system or effect. One can, for instance, use the jittering effects from the spacecraft or not, or employ differently modeled spectral analysis and so on.

SOPHISM, then, is compelled to deal with all the imaging, polarimetric, and spectroscopic effects plus other additional ones that may compromise the SO/PHI behavior. Among them, we may mention firstly some effects induced by the platform, e.g., residual jittering and/or temperature- or stress-induced birefringence in the heat-rejection entrance windows of the two telescopes. Secondly, the special characteristics of the mission orbit, imply very reduced telemetry capabilities that require some of the scientific analysis to be made aboard the spacecraft. Instead of the raw data, the instrument is designed to downlink only solar atmospheric quantities like the three components of the magnetic field, the line-of-sight (LOS) velocity, or the continuum intensity. All this highly computer-demanding data processing should be entailed too by SOPHISM and the corresponding blocks will be described here. A block diagram of the simulator is shown in Figure 1.

Thanks to the SOPHISM modularity, we can focus its description to simulations of the HRT when all modules are used. The Sun is modeled through MHD simulations that provide realistic data arrays specifying the physical quantities prevailing in the solar atmosphere in 3D. These MHD data are used to synthesize the Stokes profile spectrum emerging from the solar surface by solving the radiative transfer equation with codes like SPINOR (reference) or SIR (reference). A spatial replication of the Stokes parameter images in order to simulate a larger field of view (FOV) is used as input to the simulator. A first block is in charge of preparing these input data. It takes a time series of MHD simulations very finely sampled in time or interpolated among coarsely sampled ones in order to simulate a “continuously” varying Sun that is observed by a finite exposure time detector. With this concept, a simulated observed frame (or image) will be the result of having integrated a given number of “sub-frames” in the detector. The second block in SOPHISM deals with the effects of spacecraft jittering that “shakes” the solar image (every sub-frame). The frequency content of the jitter is parametrized into X and Y motions that are applied to the input images. The (on-line) attenuation spectrum from the image stabilization system (ISS) in SO/PHI is applied in this very module where the motions are corrected. Since the ISS only acts to the HRT, its attenuation is skipped when a simulation of the FDT is being carried out. The effects of all the optical elements before the polarization analysis are dealt with in a third module. The theoretical optical transfer function (OTF, Fourier transform of the point spread function —

PSF) of the telescope can be modified according to possible low-order aberrations (defocus, coma, astigmatism, spherical, etc.) by means of their corresponding Zernike polynomial coefficients. A fourth block cares about the polarimetric properties of the system prior to the polarization modulator through a given Mueller matrix and then with the polarimetric modulation scheme chosen for the two LCVRs. An option for dual-beam polarimetry is also included in SOPHISM. A fifth and a sixth blocks account for the spectroscopy by simulating both the pre-filter and the etalon. Since the SO/PHI etalon is on a focal plane (because the configuration is telecentric), the effects of pupil apodization are dealt with in a seventh block. (Further account of the different transmissions across the etalon can be taken into account but are not included yet. This would imply a different PSF per pixel.) The detector, including the effects of a shutter, is simulated in an eighth block. The exposure time, together with the characteristic time for the LCVRs to change the polarization state, drives the selection of frames that are useful or need be discarded in order not to introduce spurious polarization effects in the measurements. Photonic noise can also be introduced in this block. Dark-current and flat-field corrections are carried out. Accumulation (N_a different images of the same polarization state are added up in order to increase the S/N ratio) is made in a ninth block. Polarimetric demodulation is performed in the tenth block. A subsequent data analysis block is in charge of carrying out an inversion of the radiative transfer equation (RTE) in order to translate the raw data into solar physical quantities of interest that are later compressed in the last (twelfth) block and driven to the spacecraft massive data storage prior to being sent to Earth.

3. IMPLEMENTATION

The simulator is completely coded in IDL. Although this probably hinders its performance speed, it allows an easy management of graphical representation of results and a comfortable work with data arrays of any number of dimensions.

All the image data handled by SOPHISM are written in FITS format, hence allowing for metadata to be included in their headers. All the information about processes and variables needed to characterize the data can be included in those headers. Along with the FITS files, the various by-product outputs from the simulator are stored in *save* IDL format. From the initial settings of the simulation through the results of operations, calculated curves, and/or ad-hoc generated images are stored. Some of these data are used in posterior processes and most of them are employed in a final report of the simulation.

To help the organization and modification of the code, SOPHISM is made up of several individual blocks or modules. Each one of them performs a series of operations related to a particular aspect of the instrument or its environment. The operations within every module are taken on as independent of the rest as possible. This way, modules can be enabled or disabled in each simulator run in order to take into account one effect or the other, or to focus only on a few of them. Modularity also facilitates the implementation of new effects or sub-systems (modules) without perturbing too much the global scheme.

The front-end of the simulator is a graphical user interface (GUI) where the different modules are set or not and where required variables or parameters are specified. Paths and filenames of inputs and outputs are also given in this GUI. An example of one the GUI tabs is shown in Figure 2. As an alter-

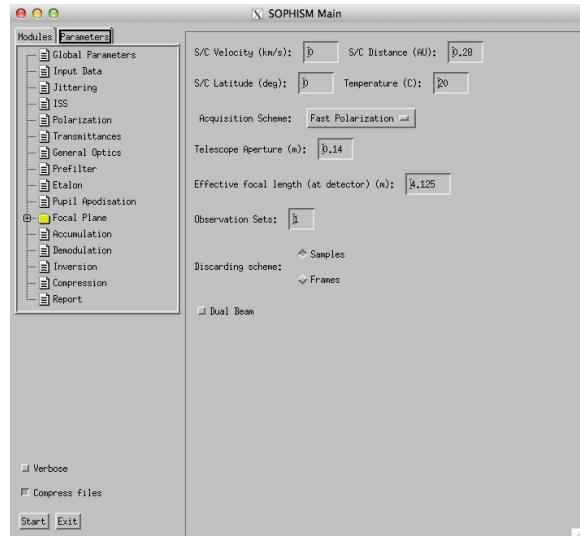


FIG. 2.— An example of the SOPHISM graphical user interface. Left area corresponds to the tab selections for enabling the different modules – in the Modules part – and for setting the input variables for each module – the Parameters part –. Also, the Verbose, Compress files and the Start/Exit buttons are found in this area. Right area displays the settings associated to the given tab and the fields to enter the user defined values. In this case, the tab shown corresponds to the Global Parameters. See text for further details. native way to start the simulations, an ASCII file—following a provided template— may be used to enter all the necessary settings and run SOPHISM directly without opening the GUI.

All the routines are commented at the beginning of their main bodies. The purpose of the routine, some of its operations, input and output variables, super- and sub-routines, and history of versions are included in those headers.

3.1. Settings

The front-end GUI has two tabs to fully characterize a given run of the simulator: one for enabling modules (SOPHISM modularity allows enabling or disabling several modules in each specific run) along with entering input/output paths and filenames, and another for selecting values for the simulation parameters.

The simulator is designed to account for all the effects that will take place during a SO/PHI observation. Thus, every module should be enabled in a simulation run for a proper description of an observation acquisition for the instrument. However, the option of disabling specific modules allows for tests and/or focusing on particular aspects of the instrument without other effects masking or perturbing the one requested for analysis.

Modularity helps as well when running several simulations that only differ slightly (e.g. to test the impact of different values in a particular setting) or in the event of using observations obtained from a different instrument as input data for SOPHISM simulations (it would be needed to disable the modules that correspond to effects already present on the data). Finally, taking into account the generalization and application of the simulator to other instruments, the enabling and disabling of modules is a required option given that not every instrument will have all the elements of SO/PHI, e.g. a polarization modulator.

In the filenames area, not only the path and names are specified but also the loading option for the settings file, in ASCII or IDL *save*.

3.1.1. Global parameters

The parameters to be specified for each module can be found in the clickable options of the GUI `Parameters` tab. A few of them affect the general behavior of the simulation and are included in the so-called `Global parameters` option. These global parameters are the spacecraft velocity and distance relative to the Sun, its latitude about the ecliptic, a temperature characterizing the global operation status of the instrument, the frame acquisition scheme, the discarding scheme, the telescope aperture, the dual beam option, the verbose option, and the compress files option. The first three parameters are important because the orbit of our spacecraft is such that they will dramatically change for the different operation epochs. The relative velocity excursions will be as large as $\pm 25 \text{ km s}^{-1}$. The corresponding wavelength shifts of the spectral line need to be compensated for. The distance to the Sun determines the spatial resolution and/or the size of the solar disk. The latitude over the ecliptic may be useful when polar magnetic field simulations are available because it modifies the viewing angle of the solar poles. The global temperature is not used yet but is prepared for future updates where, for instance, the effect of a stress-induced birefringence of the heat rejection entrance windows can be modeled. The telescope aperture needs no comment. When the verbose option is on, each module will produce some example outputs on the screen like generated profiles, resulting images, some calculations, module progress, etc. The compress option enables compression of the FITS output files by using the `writelfits` command in IDL in order to save some disk space.

Since changes in the LCVR birefringence and tuning of the etalon are not instantaneous, there will always be some times when a continuously operated detector is recording mixed states in either polarization or wavelength. This is the case of the SO/PHI CMOS detector, which has a rolling shutter. Whether these mixed states are useful depend on the scientific requirements that translate into polarimetric or spectroscopic accuracy. If there are useless data, they must be discarded. To simulate the data discarding process, SOPHISM follows the two possibilities that are considered so far for the real detector, namely, full frame discard and synchronized data acquisition. They correspond to a continuously operated detector (like the one in SO/PHI with a rolling shutter) or a triggered detector (that may represent better detectors with a mechanical shutter), respectively. When the frame discard option is chosen, SOPHISM calculates the closest number of frames (rounding up) that fall during the changing time of either the LCVRs or the etalon. These frames are discarded and storage is resumed when the system is again on a stable state. If the synchronization option is selected, SOPHISM calculates the number of sub-frames that are recorded during the changing time. Only these LCVR or etalon latency sub-frames are discarded by triggering the detector right after the changing time is over.

The dual beam option is available for those instruments carrying out dual beam polarimetry. These instruments have a polarizing beam splitter as a polarization analyzer, so that two orthogonal polarized states can be recorded at the same time. The assumption in SOPHISM is that each state is recorded by a single camera. All modulation steps are done simultaneously for both cameras and the demodulated Stokes parameters are added at the end, getting rid of most motion-induced crosstalk between Stokes parameters.

3.1.2. Modules

Twelve modules are currently included in the simulator. They are described in the next paragraphs, arranged according to the conceptual order followed by light through its path across the system. The solar Stokes spectrum images and their possible evolution are modeled outside SOPHISM. The input data to SOPHISM build upon these external simulations.

Input — SOPHISM accepts arrays of data in FITS format, structured in four dimensions, namely, wavelength, Stokes parameter, space horizontal (X), and space vertical (Y). As explained in Sect. 2, the arrays are built from MHD simulations and Stokes parameter synthesis codes. These 4D hypercubes can be arranged and processed in the code in order to fully represent the specific simulation. Data from actual instruments can also be entered, with the same dimensional structure stated above.

The code is ready to replicate the hypercubes spatially in order to simulate a larger solar area, provided the input data have periodic boundary conditions. One can also resample the input data in both the X and Y dimensions to the detector plate scale (specified) in the detector module.

Regarding temporal evolution, there are three different ways to prepare and work with input data. The first possibility is to work with a single time snapshot. Despite the limitations this mode has in the simulator (various modules require a time series, real or artificial, to simulate their behavior) it may still be useful for some particular tests without the need to produce and store bunches of data. The second temporal option is to create an artificial time series by replicating a single hypercube in order to simulate a series of frames with their corresponding sub-frames. This option would correspond to a time series where the solar scene is ‘frozen’, without temporal evolution. The number of sub-frames needed to be replicated is calculated by the simulator from the total observation time and the temporal sampling assign to the sub-frames generated. A third option is possible when an external simulated time series is used. The simulator can interpolate in time such a series from its original temporal sampling (specified by the user) to a requested one. This third option would be the most representative of actual observations, with a solar scene changing in time.

Jittering and image stabilization — Any image motion effect produced by either seeing (for ground-based instruments) or residual jittering (for space-borne instruments) and its attenuation by the action of a real time image stabilization system (ISS) are modeled in this module. If jittering is set on, image X and Y displacements are contained in a $(2, N_s)$ array, N_s being the total number of sub-frames in the simulation. Shifts are applied to every sub-frame in the Fourier spatial domain by use of the `fft_shift` IDL routine.

Three are the options for calculating the spatial displacements. First, one can use a zero-mean, white noise (flat frequency spectrum) random time series of a specified rms in arcseconds. Second, from the former jittering data, one can filter out a range of frequencies (the others are set to zero) or all the frequencies above or below a given threshold. Third, a shift power spectrum mimicking that from a real source (e.g., that from the Hinode spacecraft; Katsukawa et al. ...). The latter is obtained by convolving (multiplying in the Fourier domain) the initial white noise time series with that coming from the real source. The frequency spectrum manipulations in options two and three imply a change in the specified rms. A subsequent re-normalization is carried out within this mod-

ule.

Conceptually speaking, image stabilization should be dealt with in a separate module. Nevertheless, since only one attenuation spectrum is currently used in SOPHISM, the real time corrections that the SO/PHI correlation tracker exerts over the live image are also included in the same module as the perturbations. A set of discrete frequency attenuation power values, that are stored in the simulator routines, are fit to a power law, according to the expected behavior of the ISS. The jittering spectrum is thus divided by that attenuation spectrum (wherever is different from zero, of course). Since the ISS acts on the HRT M2 tip-tilt mirror and is off when the FDT is working, image stabilization is only used for HRT operation simulation.

The telescope(s)— The transformations on a plane parallel wavefront of a given wavelength acted by the telescope (either the HRT or the FDT) are included in this module. The OTF does not only include diffraction by a circular aperture but also aberrations are considered through a Zernike polynomial description. The simulator allows two ways to enter the aberrations:

- Directly providing the corresponding coefficients of low-order Zernike polynomial terms. In this way, the user may set values for Tip-Tilt, Defocus, Astigmatism, Coma and Spherical aberrations. The defocus term has an alternative input unit due to the more usual way of stating it as distance between the focal plane and the detector. Therefore, the user can assign a value to this defocus distance and the code will calculate the equivalent coefficient for the Zernike term, based on the telescope aperture, focal length and wavelength.
- Loading a collection of Zernike coefficients in an ASCII file. The file would contain up to 45 rows of coefficients, corresponding to the first 44 terms of the Zernike expansion (the zero term corresponds to the pupil mask).

Both ways of providing the wavefront can be used at the same time, so the loaded collection of coefficients will be altered by the user-specified low-order values. The final wavefront arising from these operations can be re-normalized to a given rms value.

Once the wavefront is ready, the pupil function is prepared with amplitude unity inside the pupil and zero outside, and with the wavefront in the phase term. The OTF is obtained as the auto-correlation of that function and a convolution with the data is performed.

When pupil apodization is considered (the case for SO/PHI, with the etalon in telecentric mounting), the calculations are more complex. Since there is an un-homogeneous illumination from the pupil due to the converging light cone, the amplitude term is now calculated from the etalon transmitted intensity. Therefore, the amplitudes are a function of the pupil space and of wavelength, and this goes into the OTF. Moreover, the passing through the etalon perturbs the phase of the incoming light. So in this description of the OTF, the phase term derived from the wavefront and aberrations calculated above has an extra term from the etalon perturbation. The OTF has now the usual dependence in pupil space (complicated in this description because the amplitudes are different along the pupil) and the dependence in wavelength, which is actually twofold. One part is the wavelength dependence

proper, because of the spectral profile from the etalon. The other part refers to the different spectral positions selected, each of them having a different spectral profile. Finally, when dealing with pupil apodization mode, the OTFs are stored because the convolution is not performed here but in the Pupil Apodization module.

Polarization— As stated above, SO/PHI uses a pair of nematic LCVRs and a linear polarizer for the polarimetric analysis.

This module deals not only with the desired polarization modulation, introduced by operating the LCVRs, but also with the Mueller matrix of the system which will modify the given polarization of the input data.

The Mueller matrix of the system can be entered as a combination of optical elements, mirrors and retarders, or as an ideal identity matrix. It may also have variations with wavelength or along the FOV. The Mueller matrixes of the LCVRs are calculated from the four retardances for the LCVRs introduced by the user. Like the system matrix, the LCVRs parameters can depend on wavelength and FOV. The modulation matrix then is calculated from the combination of both LCVRs, the matrix of the system and the linear polarizer. The possible dependences in wavelength and FOV of system and LCVRs parameters will result in arrays of modulation matrixes to be applied for each wavelength or pixel in the FOV.

Demodulation matrixes will be obtained by directly inverting the modulation ones. In the event of having wavelength or FOV dependence, a second demodulation matrix will be calculated by averaging. This will be applied in the Demodulation module to produce two sets of data, ones demodulated with the complete matrixes and another with the average.

For both the system matrix and the LCVRs there is an option in SOPHISM that allows the introduction of uncertainties. In the case of the system matrix, the uncertainty is created by generating an artificial retarder with the settings specified by the user for the retardances, orientation and amplitude. This artificial element will enter last in the calculation of the Mueller matrix. With the LCVRs, the uncertainties are applied directly on the settings selected for them, i.e. modifying the retardances, the orientations or amplitudes of each LCVR. When uncertainties are introduced, the modulation matrix is done twice: with the 'nominal' settings (without neither errors in the LCVRs parameters nor artificial retarder for the system) and with the actual ones, including all the uncertainties. The data will be modulated with the actual matrixes, not the nominal ones. But demodulation matrixes of both actual and nominal will be produced and used in the Demodulation module to obtain the real demodulated data and data demodulated with the expected, incorrect, matrix.

The change of the LCVRs when applying voltage is not instantaneous. It takes them time to react to the voltage and move to another state. During this time, the modulation matrixes are changing and data observed at those times should not be used. Those 'latence' times can be input by the user, one for each of the four modulation states. Here it is considered four to be the maximum number of states (it could be two in a longitudinal mode) and that both crystals change at the same time, so only one time (the longest) is needed per state change. The two modes of representing this latence situation in the simulation were commented above. Its application in this module consists in throwing away, not considering, the appropriate number of subframes that fall in a latence time.

Some acquisition schemes of the observations required that the modulation cycle is repeated a number of times before tuning the etalon to a different wavelength. The option for such scheme is included in SOPHISM and this module will automatically repeat the modulation scheme the given number of times.

If the dual-beam option was selected for the simulation, two sets of data will be produced here, with orthogonal modulations respect to each other. The demodulation matrixes will be inverted for both sets of modulations and the two datasets carried over for the whole simulation, going through the same processes except at the detector (see `Detector` module description).

Pre-filter — When using an etalon for spectroscopic analysis, there is an extra element needed: a pre-filter. This necessity comes from the periodicity of the transmission peaks produced by the Fabry-Pérot, which means that one will get light transmitted through the etalon not only at the required wavelength, but also at periodic wavelengths off the desired one. The distance between two consecutive transmission peaks is the free spectral range. The aim of the pre-filter is to block all those other transmissions while allowing the selected one to go through as unperturbed as possible.

In SOPHISM, this pre-filter element may be included from a theoretical approach or from actual measurements. A transmission curve may be provided in ASCII format, comprised of two columns: the first one with wavelengths and the second one with the transmissions at such wavelengths, normalized to the maximum. If an actual curve is not provided, the code can calculate a theoretical pre-filter transmission, given some essential parameters, by means of a modified Lorentzian curve:

$$T = \left\{ 1 + \left(\frac{\lambda - \lambda_c}{FWHM/2} \right)^{2 \cdot n_c} \right\}^{-1} \quad (1)$$

where T is the transmission, λ_c is the central wavelength of the pre-filter, $FWHM$ is the full width half maximum and n_c its number of cavities.

The spectral PSF for the instrument will be the combination of the etalon transmission and this pre-filter transmission.

Etalon — The spectral analysis in SO/PHI and SOPHISM is done with a Fabry-Perot etalon. More precisely, a LiNbO_3 solid etalon. The desired wavelength is tuned by applying voltages to the etalon which, in turn, modifies its refractive index. Refractive index that also depends on wavelength, apart from temperature. The strategy for the simulation is, once set the refractive index with the temperature and voltage, the transmission order and wavelength are calculated. Then, an iterative process begins re-calculating the refractive index at this last wavelength, the transmission order and the new wavelength derived. This is repeated until the refractive index does not vary. This iterative process is needed because the dependence of the refractive index with wavelength. The refractive index is slightly different for each new calculated wavelength, which itself is slightly different for each new refractive index. Up to a point where there is no measurable difference.

SOPHISM admits two ways of providing the spectral positions: specifying the voltage applied to the etalon or the wavelengths to tune the etalon. Since the input data is discrete in wavelength and not every wavelength is sampled, when requiring a specific wavelength (or voltage), the code will do an interpolation with the two nearest wavelengths sampled to obtain the one asked.

There is an option to introduce a ‘jittering’ in the voltage tuning of the etalon. This option represents an uncertainty in the voltages supplied to the etalon. The etalon then will tune to slightly different spectral positions than expected. The simulator modifies the voltage with a random deviation with the given rms.

The usual characteristics for describing an etalon have to be provided, like thickness, reflectivity and fabrication finesse. Also, the F-number at the etalon, which in essence represents the mounting of the etalon, a low F-number implying a fast convergence (i.e. higher angles of incoming light convergence) and higher F-numbers tending to a collimated etalon approach. In the case of SO/PHI, the etalon is in a telecentric mounting, close to a focal plane. Because of this, each point from the FOV incoming light goes through a different point of the etalon. Moreover, the light is collimated at the pupil and converges in a cone on a point of the etalon. This means that each pixel in the output image comes from an integration of a light cone over the etalon surface. A cone means that the light is formed by a composition of an array of angles, and different incident angles correspond to different transmitted wavelengths, intensities and phases. Because the integration of all the converging angles, the transmitted light has a blueshift with respect to the desired wavelength (which would only be correct for the central light ray of the cone). The cone is the same for every point of the image, so it only needs to be calculated once and applied to the whole etalon surface and input data. As long as the etalon has a constant thickness.

SOPHISM allows a simple treatment of the etalon when the pupil apodization module is not enabled. Then, an average of the light cone is calculated without any more 2D consideration. When the pupil apodization is enabled, a full 2D treatment of the incoming light through the full pupil is calculated.

The telecentric configuration has another drawback in the etalon surface roughness. Because the light cone covers only small areas of the etalon surface for each point of the image plane and the etalon surface is not perfectly flat, each point will ‘see’ a different etalon thickness. Different thickness implies different wavelength transmission which means that the output image will have a slightly shifted profiles from pixel to pixel. Moreover, the transmitted intensities and phases also change from etalon point to etalon point now. This will have an important impact for the OTF calculation, yielding a space-variant PSF.

Like in the case of the LCVRs, the etalon tuning from one selected spectral position to the next is not instantaneous. In this case, the time needed to change between positions, the tuning speed, is given by the limitation not to surpass 1500 Volts per second. This is considered as a safety precaution not to affect too negatively the lifetime of the element. This value can be changed in SOPHISM. With this tuning speed and the voltage sensitivity of the etalon, the time needed per miliangstrom shift is obtained. Again, like with LCVRs, the subframes that fall into the etalon latency time are discarded. Depending also on the discarding scheme selected. It is considered that when both LCVRs and etalon have to change states, they are being operated and the same time, so only the longest between both latency times is considered for those occasions.

Pupil apodization — As commented along the descriptions of the Telescope and Etalon modules, the pupil apodization is a well-known effect **REFERENCIA**, consequence of placing the etalon in a telecentric mounting. Because of this, the light

coming from the pupil converges in a cone on the etalon surface. The light integrated at each point of the etalon is a combination of incident angles of the cone, implying a variation in wavelength transmission. This has, in turn, repercussions on the OTF.

The thick of the simulation effects related to the pupil apodization is done in the Telescope and Etalon modules. This module consists in the convolution with the data. For each selected spectral position, the data is convolved with the OTF at each wavelength (the OTF changes with wavelength). Then, it is weighted with the average etalon transmission at such wavelength and integrated for all wavelengths, for each spectral position.

The detector — The simulation of the detector in SOPHISM implies several levels, from purely instrumental (characterization of the detector regarding response to incoming light, noise level, etc) to representing different ways of data recording, passing by conversions of units and effects of the whole instrument.

Several units conversions take place in this module, necessary for some calculations performed (e.g. photon noise, for which the number of photons is needed). The input data is supposed to be in energy flux units ($\text{erg/s/cm}^2/\text{strad/cm}$) and is converted to photons according to the telescope aperture, plate scale, wavelength, FWHM of the spectral profile and the exposure time selected. Then, photons are converted to electrons through the quantum efficiency given and those into counts, or digital numbers, which are the product of the detector once read out.

Although not properly an intrinsic attribute of the detector, the transmittance of the system is treated in this module. Presently, this aspect of the instrument is treated as a single value that would summarize the behavior of the whole system. Thus, the intensity of the incoming light would have been attenuated by this quantity and it is reflected in the detector simulation as the level of the light reaching it.

Detectors present a so-called dark current, i.e. an average level of electrons with a random pattern around the FOV just by functioning. The mean level depends on the temperature of the detector and on the radiation dose that it is receiving. The latter is important for spacecraft instruments and both will change in SO/PHI with the orbit and with lifetime. The level of dark current in an observation is also a function of the exposure time.

When performing observations with actual instruments, it is always needed to take ‘flat-field’ images. These data shows the different response of the detector to an homogeneous illumination covering the whole FOV. SOPHISM covers this response in two separate elements: gain table and flat. The gain table part comprises the different reaction of every pixel in the detector to the same incoming light and is a random realization over the FOV with a user-defined scaling factor. The flat section is more a large scale behavior. It would represent intensity gradients over the whole detector and is generated in the simulator by a random realization over the FOV smoothed with a surface fit. In the random realization, the simulator admits an option to indicate the range (maximum to minimum) of the intensity variation over the FOV to be achieved. It also allows the user to specify if a different flat must be produced for each wavelength and/or modulation state, given that the tuning of the etalon or LCVRs may influence on the incoming light.

On top of the above flat-field, there are interference fringes.

Interference fringes are a common problem arising because of the mere presence of reflective optical surfaces and their impact on data must be studied for every instrument. SOPHISM includes an option to generate fringes in the data observed by the detector. A simple version of random fringes with given amplitude and angle is available, with the possibility of changing with different LCVRs and/or etalon states. A more accurate depiction of actual fringes generation is also available, based in the reflectivity, thickness and refractive index of the element producing the fringes and in the wavelength and incidence angle of the incoming light.

There are two more noises that can be included in the simulation of the detector: photon noise and readout noise. The former is an intrinsic property of observations because of the nature of photons (or the electron generation in the detector) and depends on the number of photons arriving at the detector. It is calculated as a Poisson distribution independently for each wavelength and modulation state. The readout noise is a consequence of the reading/writing in the detector when moving the charges and is represented as a normal distribution, with a standard deviation provided by the user.

All the effects of the detector described above (dark current, gain table, flat, fringes, photon noise, readout noise) may be simulated from scratch in SOPHISM or can be loaded from an appropriate IDL `save` file containing the corresponding arrays.

Now, the actual simulation of the data recording in the detector will mainly depend on the kind of shutter selected, snapshot or rolling. First of all, since the input data is discrete in time, each exposure will consist on the summation of a number of subframes given by the relation between the selected exposure time and the subframes temporal resolution. For example, in the case of input data with a cadence of 5 ms and an exposure time of 20 ms in the detector, each frame would be the addition of 4 subframes. The way to do this integration of subframes is what depends on the shutter:

- **Snapshot:** This kind of shutter is a full FOV shutter. It exposes the whole detector at once. The representation of this shutter in SOPHISM consists in a direct addition of the subframes that fall into the exposure time. If the exposure time is lower than the readout time, the remaining subframes until the next frame starts will not be considered.
- **Rolling:** The rolling shutter exposes one (or various) row of the detector during an exposure time. In reality, this means that each row will ‘see’ a slightly different scene, due to solar evolution, pointing instability and any other change happening. The way to consider this in SOPHISM is by dividing the would-be final frame in blocks of rows. The size of each of these blocks is defined by the relation between the number of rows in the data and the number of sub-frames to form a frame. Each block will then be constituted by a different sub-frame, with a mixing of sub-frame in the rows around the change of block, simulating a transition/evolution into the following sub-frame.

In the case that the dual-beam option has been selected, new realizations of every random variable (e.g. photon noise, readout noise, etc) will be made for the second set of data. In this way, a second independent detector is represented. Likewise, the transmittance of the system is halved which would correspond to a perfect beamsplitter dividing the incoming light in

two beams, one for each detector.

Image accumulation — Image accumulation consists in adding together several exposures of the same solar scene in the same instrument state (spectral and polarimetrically) so the S/N ratio increases. This accumulation can be done in several ways and orders, some of them pertaining to the selection of an acquisition scheme and repetition of observation cycles, in modulation, wavelength or both. But finally, all the schemes result in a summation of exposures that correspond to a common state.

The adding of the accumulations entered in the Number of Accumulations parameter is performed in this module. The process itself consists in the simple addition of the given number of exposures, in the appropriate spectral and modulation index. The module also re-arranges all the individual data files that have come out of previous modules into a single data array and file per observation set.

In the end, the output of this module are the final raw data obtained from the instrument. This is the final step of the ‘hardware’ observation and simulation, before any reduction or software treatment.

Polarimetric demodulation — This module belongs already to the data pipeline, or ‘software’ part of the simulator (meaning that it represents how the actual instrument will reduce and treat the data before sending it by telemetry).

The demodulation is a process usually performed by the observer after acquiring the data. The later is modulated in the instrument when observing and the observer produces the Stokes parameters by post-facto calculations making use of demodulation matrixes obtained for the particular combination of telescope and instrument. These demodulation matrixes are obtained from calibrations made for the instrument. In the case of the SO/PHI instrument, the process of demodulation will be fulfilled autonomously on board.

The aim of the module is straightforward: to demodulate the data, to obtain the Stokes parameters from the input modulated at the Polarization module. To that end, the demodulation matrixes calculated and stored during the Polarization module are restored in here. The input data of the module is read and the demodulation applied, producing a demodulated data array comprised of all the spectral positions selected and the Stokes vector corresponding to the modulation, be it a full Stokes modulation or a longitudinal one.

In SOPHISM, the demodulation is always processed with the invert of the modulation matrix, as it should be. The result of this would be a perfect demodulation. But in reality the exact demodulation matrix is not always known. There are uncertainties in the retardances induced by the modulator, in its dependencies with FOV and wavelength, in the Mueller matrix of the system, etc. In order to reproduce this uncertainties problem, the simulator calculates two sets of demodulated data whenever the modulation is not the simplest case (i.e. if the modulation includes dependencies in FOV and/or wavelength, or if the option for unknown errors in the Mueller matrix or in the retardances of the LCVRs has been selected). One of the output sets is demodulated with the actual demodulation matrix/es coming from the inversion of the modulation one/s. The other set is the result of demodulating with a single simplified, or expected, version of that matrix. If the modulation has FOV or wavelength dependencies, the ‘simple’ demodulation matrix will be an average of all the demodula-

tion matrixes for each pixel or wavelength. If the uncertainty comes from the errors option in the modulation, the demodulation matrix will be the invert of the expected modulation, without taking into account the errors. In this way, the user will have directly as output the correctly demodulated data and data demodulated with the ‘reality’-expected modulation, so both sets can be compared.

In the case of a dual-beam simulation, the two series of demodulated data from the orthogonal modulations are added together at the end of the module, producing already the final dual-beam output.

The demodulation module presents one option for the user to enable: Ad-hoc Crosstalk Correction. This option reproduces calculations done commonly by the researchers during data reduction, and is intended to correct crosstalk between Stokes parameters when demodulating data. These crosstalks originate from two sources: from an unaccurate demodulation (e.g. because of uncertainties in the modulation parameters which in turn cause that the expected demodulation matrix to apply is not the actual one) or from actual changes in the observed object (e.g. because of pointing instabilities or because of solar evolution during the modulation, making the solar scene different among the recorded modulation states). The consequence of the crosstalks is that any of the Stokes parameters will have a remaining contribution from the other three parameters, creating spurious signals that will affect the magnetic field information retrieval. The ad-hoc correction is done in two steps:

- Stokes I into Q, U and V crosstalk: Stokes I is removed from Stokes Q, U and V at every spectral position. Stokes I is normalized by the ratio at continuum between Stokes I and the corresponding Stokes parameter to be corrected before it is subtracted.
- Stokes V into Q and U crosstalk: linear fits are calculated between Stokes V and each of Stokes Q and U respectively, at each spectral position. Then the slope of the fit is used as normalization for Stokes V before subtracting it from the corresponding Stokes parameter.

When the Ad-hoc Crosstalk Correction is engaged, a new set of output data will be obtained with the correction applied so the user may compare corrected vs uncorrected results.

Inversion of the RTE — Limitations on the telemetry advised to come up with solutions to reduce the data download from SO/PHI while keeping the necessary science data products at appropriate levels of FOV size and quality (i.e. without too much lossy compression or binning/trimming).

For this reason, SO/PHI will perform on-board inversion of the reduced data. Thus, the data for download will be cut down to continuum intensity image and velocity and magnetic field vector (strength, inclination and azimuth) maps from the original six wavelengths times four Stokes parameters data.

SOPHISM reproduces the inversion work that will take place in the instrument. Both the instrument and the simulator use MILOS code **REFERENCIA** for making a Milne-Eddington inversion. **First, the classical estimates (center of gravity) are calculated. Then, those results are used as initial guess of the MILOS inversion.** Only the pixels with signal above a sigma threshold provided by the user are inverted. For the pixels that enter the inverter, a maximum number of iterations is set. It is also necessary to specify the position of the continuum in the spectral positions array. This

will also be important in the actual instrument due to the velocities in the orbit that will shift the spectral line too close to the extremes of the pre-filter. In those cases, the observation plan will change the sign of the continuum position, from the blue side of the line center to the red side, or viceversa.

The inversion is performed for each data output from the Demodulation module, i.e. if the modulation had wavelength of FOV dependencies or random errors, which produces extra outputs, all those data will be entered in the inversion.

Report — This final module does not correspond to any part or effect of the instrument and does not affect the data in any way. It is a summary of the simulation run performed, compiled and presented in a PDF file.

The report is prepared in LaTeX, with eps figures, and automatically compiled to produce the PDF. It is divided in the different modules selected for the simulation and contains the values for the main settings that the user has input. Plots and figures pertaining to the calculations done in each module (e.g. spectral PSF profile, dark current frame, etc) are depicted at the corresponding steps in the report. Example data outputs from each module are also included so the impact of the operations carried out in each module can be seen directly.

By default, the report is prepared taking into account every module enabled for the simulation, but only those. However, a specific set of modules to be included in the report may be selected, disregarding whether they have been enabled for the current run or not. This option is useful when starting simulations from a particular module using input data from some simulation step of a former run (due to, for example, two sets of similar simulations with a modification of a specific parameter from one to the other to test its effect and thus, the modules previous to that parameter do not need to be run again). Then, the modules that were run previously and from where the input data came in the new simulation can appear in the report.

4. EXAMPLES

The following subsections show examples of SOPHISM at work which allows to get a more visual impression of the simulator operations.

To illustrate the results that can be obtained with the simulator, the first section presents image outputs of the modules during a simulation run. The second section contains an example of a possible underperformance in an instrument and tests of two ways to overcome it. A third section presents an alternative use of SOPHISM for night observations to demonstrate the versatility of the simulator.

4.1. Modules

As stated previously, each SOPHISM module performs a series of calculations and operations, and applies them to the module's input data (except when Pupil Apodization is enabled because Filtergraph and Telescope operations are then directly applied in the Pupil Apodization module). Thus, looking at the data outputs from each module as a sequence tells the story of a simulation run, and hence of the path and elements that the incoming light has gone through.

An example of such a sequence of output data can be seen in Fig. 3. It shows intensity images (Stokes I or modulated images where corresponding) from the different modules' output of a whole simulation run. These example data are shown for a spectral position at the minimum of the Fe I 6173 Å line.

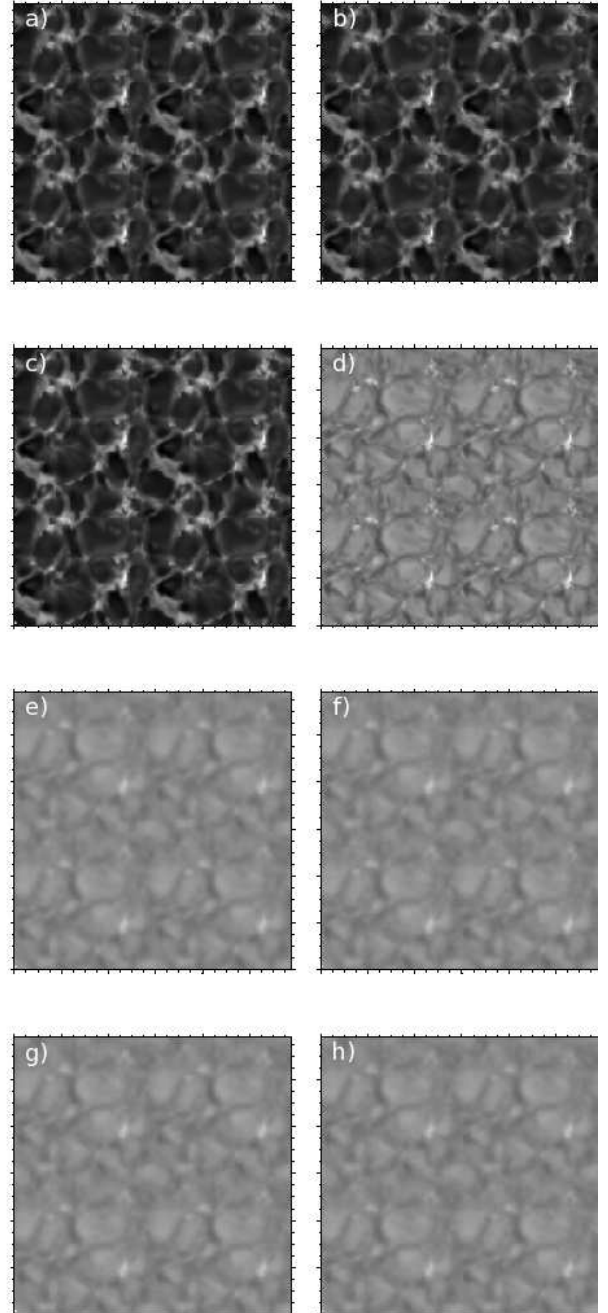


FIG. 3.— Left to right and top to bottom, outputs from: input module (a), jittering (b), modulation (c), filtergraph (d), OTF (e), FPA (f), accumulation (g), demodulation (h). All of the images are scaled to the same range. Note the four-times repeated pattern in each image, consequence of a 2x2 spatial replication in the input module.

For this typical simulation, the Pupil Apodization module was disabled. This was so just to obtain output data from both Filtergraph and Telescope modules, as it was commented before that with Pupil Apodization no independent output data from these modules is produced. In this way, it can be seen more clearly the effect of the spectral selection and of the optical aberrations on the data.

Images a) and b) look the same because from one to the other only a small shift, consequence of jittering, has been applied. Image c) is the output from modulating, and presents some differences in intensities due to the linear combination

of the four Stokes parameters. But it is image d) that shows a strong difference with the three former ones. Here, the data has passed the Filtergraph module. The output image is the result of the convolution with the transmission profile of the prefilter and etalon combination. Where before there was a specific wavelength, now there is a combination of images from different wavelengths through the spectral psf. Image e) has the effect of optical aberrations included. The degradation of quality and spatial resolution is evident in the case of this simulation. The most clear addition in image f) is the noise components added at the Detector module, like photon noise, dark, readout noise, etc. Other operations such as units conversion, exposure with rolling shutter and more have taken place but the noise components inclusion are most evident visually. Images g) and h) correspond to the Accumulation and Demodulation modules. The differences in these cases are not so noticeable although the accumulations effect have improved the S/N and the demodulation means that the image corresponds only to Stokes I and not a combination of the four Stokes parameters like g).

Following the images in Fig. 3 from a) to g), it can be seen the evolution of the observed solar scene as the incoming light goes through the instrumental elements, from the moment it enters the telescope to the final data stored. In addition, image h) shows the recovered Stokes I image once demodulated and the difference with respect to the original.

4.2. Analysis

The previous section provides a depiction of a SOPHISM running simulation, just to give a visual impression of the simulator at work and the effect that the different modules have on input data. However, the actual aim of the simulator is to analyse the behaviour of an instrument and to test it under different conditions. Likewise, SOPHISM helps examining solutions to possible problems that may arise during operations. This section shows an example of how SOPHISM can be used as an evaluation tool and test bench for overcoming such underperformances.

The simulation run consisted on testing the effect that a pointing inaccuracy perturbation has on the observations and how it affects the demodulated output data, along with solutions to correct it. The problem with jittering during observations comes because of the way modulation works. The modulation/demodulation scheme is formed by additions and subtractions of the Stokes parameters (or the modulation states to obtain the Stokes parameters, in the demodulation case). If during the observation there has been a pointing shift (e.g. because of jittering) between the modulation states, when combining those states for demodulation polarization signal will appear that was not in the original solar scene. This false polarization will have a pattern similar to granulation, since Stokes I is the largest contributor. Perturbations from the rest of Stokes parameters are also present but at lower orders of magnitude.

The simulation run for this analysis was performed with "frozen" input data, i.e. without temporal evolution of the solar scene. This way, it is clearer the impact of the jittering. When solar evolution is included, both effects get mixed because the changes in solar scene between modulation states create crosstalk in a similar way to the mentioned jittering.

The result of the simulation is presented in Fig. 4 as the output demodulated data. The four images shown in Fig. 4 correspond to the Stokes parameters at a spectral position of +160 mÅ off the line center of Fe I 6173. They are normalized

to continuum.

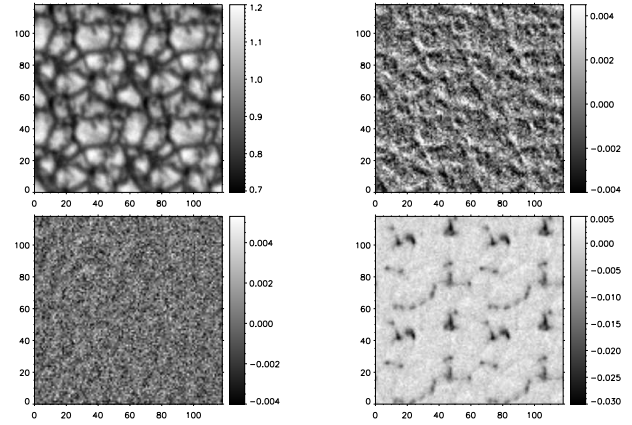


FIG. 4.— From upper left, clockwise: Stokes I/I_C , Q/I_C , U/I_C and V/I_C at +160 mÅ off line center, resulting from the simulation run with single beam design.

The impact of jittering is clearly seen in Stokes Q, yet present in every Stokes parameter. The reason because the false polarization signals are stronger in some spectral positions and Stokes parameters than in others has to do with the randomness of the jittering itself and with the order of the modulation scheme. On data analysis, results such as shown in Fig. 4 may be interpreted as signature of magnetic field, showing signal where there should be none or modifying the actual ones.

Once the behaviour of the instrument is established and the possible underperformance determined, the simulator can be used to test different solutions. Here, two approaches are presented, one based on hardware while the other is based on software corrections.

One way to overcome the jittering effect and compensate the false signals is by means of a dual-beam system, e.g. with a polarizing beam-splitter and two detectors observing simultaneously. **Because of the beam-splitter, each detector in this system records orthogonal modulation. Using the proper demodulation matrixes, data from each detector will be demodulated and added at the end, recovering the correct Stokes parameters. However, the effect of the shifts due to jittering is not ...** Another run of SOPHISM was done with the same settings as before but including the dual-beam mode. The simulation results of such an observation is presented in Fig. 5 as demodulated Stokes parameters at the same spectral position as Fig. 4.

From a simple visual impression, it is clear the improvement achieved with this scheme, being the false jittering signal not noticeable anymore. Note that, although not visible here because of the noise levels, false signal generated by the jittered Stokes Q, U and V is still present. Its amplitude is below the noise so the result here is the same as a non-jittering simulation. But if no noise were to be included in the simulation, contribution of false V, Q and U would be seen.

However, a dual-beam scheme is not always possible or advisable due to limitations in space, mass, power, etc. Thus, a different approach, based on software computation, is tried here. The idea is that, despite the randomness of the jittering – if it is actually random and has no preferent direction –, there is a predominant shift amount and direction resulting

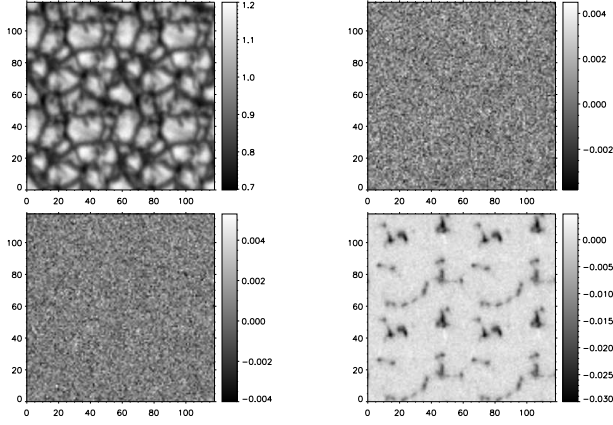


FIG. 5.— From upper left, clockwise: Stokes I/I_C , Q/I_C , U/I_C and V/I_C at $+160 \text{ m\AA}$ off line center, resulting from the simulation run with dual beam design.

from the addition of the accumulations. Although not able to completely correct the effect of the different shifts added, an alignment of the modulation states prior to demodulating may compensate the jittering impact up to some extent. To carry out this test, the same data as the simulation presented above is used. But in this case, an alignment of each modulation state after accumulation is performed. From the accumulated data recorded at the detector, the first modulation state is taken as reference and the remaining three states are aligned by cross-correlation to the first one. This is done for each spectral position.

Results from such test are shown in Fig. 6. As is already evident from comparing these images with the ones in Fig. 4, this operation does reduce the effect of jittering on the demodulated output.

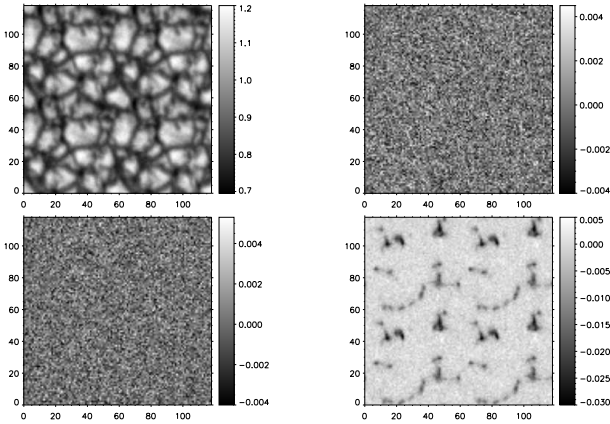


FIG. 6.— From upper left, clockwise: Stokes I/I_C , Q/I_C , U/I_C and V/I_C at $+160 \text{ m\AA}$ off line center, resulting from the simulation run with single beam design and performing an alignment of the modulated data prior to demodulation.

A simple way of checking with numbers the improvement achieved by the alignment, is seen by calculating the rms of the deviations with respect to the dual-beam case, considered as reference. Table 1 shows, at a spectral position of $+160 \text{ m\AA}$ off line center, the rms of the difference between single-beam case without alignment and dual-beam, and between single-

beam case with alignment and dual-beam.

	Q/I_C	U/I_C	V/I_C
Single	1.75459	1.10493	1.29234
Single aligned	1.18978	1.06115	1.10347

TABLE 1
TABLE OF RMS ($\times 10^{-3}$) OF THE DIFFERENCES OF THE STOKES PARAMETERS, AT $+160 \text{ m\AA}$ OFF LINE CENTER, FROM EACH SINGLE BEAM CASE, NORMAL AND ALIGNED, WITH RESPECT TO THE DUAL BEAM CASE.

The improvement is clear, specially in the Stokes parameters that were suffering more strongly from the jittering crosstalk, like Q in this case. Most of the remaining rms is due to the noise components.

Although the jittering generation starts from random, as well as the noises in the detector, several repetitions of the tests have shown similar results, always with the same behaviour. Pre-demodulation alignment helps compensating the remaining jittering in the results.

Of course, this test only aims at the crosstalk generated by jittering with a fixed solar granulation pattern. Depending on the speed of the instrument and modulation scheme, solar evolution may have a higher impact on the results than jittering. But, as for jittering, in the absence of a dual-beam system this method proves to be a useful alternative to overcome part of the jittering-generated crosstalk.

4.3. Night Observation

It has been noted above that, although initially developed to represent the SO/PHI instrument, SOPHISM has been coded with high flexibility. This allows for all kind of tests for said instrument as well as for describing different solar instruments and platforms. For example, initial contacts for representing the Visible Tunable Filter (VTF) instrument to be installed at Daniel K. Inouye Solar Telescope (DKIST) have been established.

Even further uses of the simulator may be performed with the addition of new modules or with just little modifications, like the case presented in this section. As an example of such alternative uses for SOPHISM, a simple simulation was performed considering a typical night time observation and instrumentation.

The input data was prepared to resemble a star field by placing 100 random pixels (representing point sources) with random intensities over a 512×512 pixels FOV. The maximum intensity of the star field was later scaled to the flux of Vega star (α Lyrae) received on Earth when observed with the Johnson's system V filter. This filter is centered around 5450 \AA and with a FWHM of about 880 \AA , parameters used in the simulation afterwards. Vega's flux results in $3.631 \cdot 10^{-11} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1}$ (?). The image used as input is shown in Fig. 7.

Being a simple exercise to test the simulator, no further preparation was done for the input regarding spectral or polarization information. It could be considered as an intensity-photometry analysis.

Settings for the telescope+instrument were taken from the Calar Alto 1.23 m telescope information sheet. Thus, an aperture of 1.23 m, plate scale of 0.502 arcsec/pix (from the telescope scale of 20.9 arcsec/mm and the SITE#2b detector of 24 \mu m pixel size) and a sensitivity for the CCD of $2.6 \text{ e}^-/\text{ADU}$ were used. For the combined transmission of the whole system (atmosphere and elements of the instrument), a value of

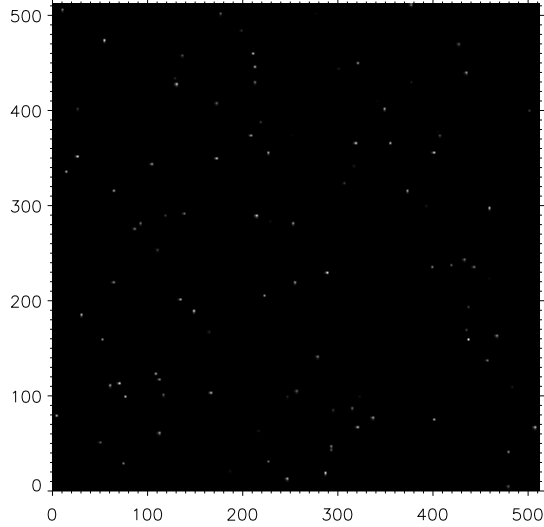


FIG. 7.— Image of input data used for night simulation. Star (as pixels, point sources) positions generated randomly, as well as intensities. The latter, scaled afterwards to Vega flux received on Earth.

0.075 was adopted. The exposure time was set to 2 ms, an approximate value to avoid saturation levels of magnitude 0 stars with the given characteristics of the telescope. The sky brightness was added to the data, following the values of ?, which show that Calar Alto's night-sky brightness is about 22 mag/arcsec² for the V filter. As a first approximation, a PSF with a FWHM of 1.5 arcsec has been considered to represent the seeing (a bit worse than the value obtained by ?), combining atmospheric and instrumental effects. This PSF has been used for the Optics Module.

The result of the simulation run is presented in Fig. 8.

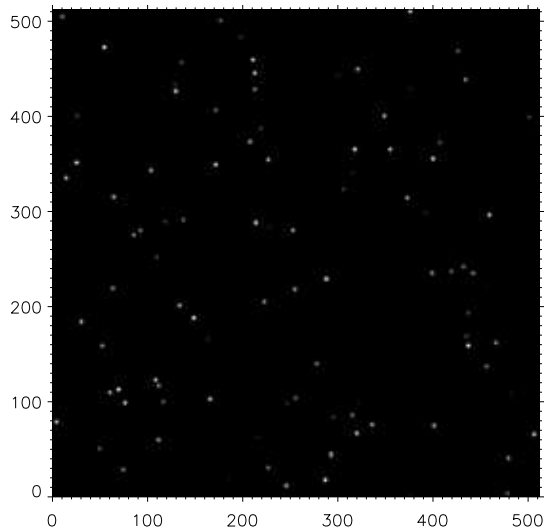


FIG. 8.— Result of simulation using Fig. 7 as input data and the settings specified in the text.

The maximum intensity in the result is of ≈ 30000 ADU, about half the saturation level of a 16 bit depth detector.

However, if the artificial star field is scaled to a maximum of a magnitude 5 star while keeping the rest of the parameters

in the same conditions as before, the result looks like Fig. 9.

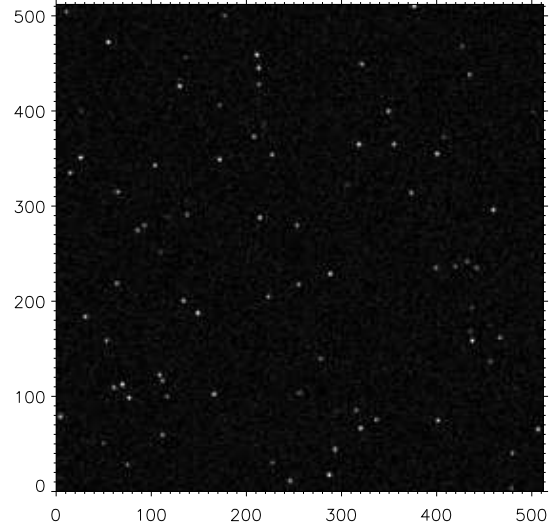


FIG. 9.— Same as Fig. 8 but scaling the maximum in the input data to a magnitude 5 star. Resulting intensities are now close to the brightness level of the night sky.

In this case of a 5-magnitude field, for the given exposure time and specifications, some of the input stars are not bright enough to be above the sky background level. The maximum intensity achieved in this latter example is two orders of magnitude lower than the former case with magnitude 0 scaling.

So with simply entering some characteristics of a night instrument and telescope, photometric results about necessary exposure time for a given magnitude star field can be derived. This limited exercise highlights the versatility of SOPHISM for several descriptions of instruments and telescopes.

5. CONCLUSIONS AND OUTLOOK

This paper describes the SOPHISM software simulator as an end-to-end simulator of instrument with platform effects and data analysis.

The examples presented have shown the flexibility allowed by the simulator

Likewise, the analysis example of a particular instrument simulated with SOPHISM, focusing on a specific aspect that could prevent from achieving optimal performance, demonstrates the usefulness of the simulator both for identifying limitations and for testing solutions to overcome the problem. In this work, it was a pointing jittering problem that created false signals onto the results. Two ways of correct it were tested. One relying on hardware which implies repercussions in the instrument design (and maybe not feasible *a posteriori*), the other involving corrections by software.

Future steps: Community Atmospheric seeing

This work has been partially funded by the Spanish Ministerio de Educación y Ciencia, through Project No. ESP2014-56169-C6, including a percentage from European FEDER funds.